-2-

Yeh *et al.*
Appl. No. 09/851,725

Amendments to the Claims

1. (Previously Amended) A protocol performance test method, comprising:
measuring performance of a multi-computer communication protocol on a single computer system, wherein said single computer system emulates a network configuration having a server and a client, and wherein measuring said performance of said multi-computer communication protocol on said single computer system includes,
      executing server code on said single computer system;
      executing client code on said single computer system; and
      measuring performance data for said single computer system.

2. (Previously Presented) The method of claim 1 wherein said server code and said client code include a number of threads, the method further comprising:
tracking an execution time for each of said threads by a processor in said single computer system; and
tracking a number of transactions completed between the execution of said server code and the execution of said client code wherein said performance data is based on said number of transactions completed over a period of time.

3. (Previously Presented) A protocol performance test method, comprising:
operating a computer system under test as a server and a client to emulate a network configuration;
executing server code on said computer system under test according to a multi-computer communication protocol;
executing client code on said computer system under test according to said multi-computer communication protocol; and
determining performance data for said computer system under test.

4. (Previously Presented) The method of claim 3 wherein said server code and said client code include a number of threads, the method further comprising:
tracking an execution time for each of said threads by a processor in said computer system under test.

5. (Previously Presented) The method of claim 4 wherein said communication protocol defines transactions between said server and said client, the method further comprising:
tracking a number of transactions completed between the execution of said server code and the execution of said client code.

6. (Original) The method of claim 5 wherein said performance data is based on said number of transactions completed over a period of time.

7. (Previously Presented) The method of claim 6 wherein said performance data is based on said number of transactions completed over said period of time modified by a scaling factor.

-3-

Yeh *et al.*
Appl. No. 09/851,725

8. (Previously Presented) The method of claim 7 wherein said scaling factor comprises a total execution time for both client and server threads divided by one of an execution time for said server threads and an execution time for said client threads.

9. (Previously Presented) A set of instructions residing in a storage medium, said set of instructions capable of being executed by a processor to implement a protocol performance test method, the method comprising:

measuring performance of a multi-computer communication protocol on a single computer system, wherein said single computer system emulates a network configuration having a server and a client, and wherein measuring said performance of said multi-computer communication protocol on said single computer system includes,

executing server code on said single computer system;

executing client code on said single computer system; and

measuring performance data for said single computer system.

10. (Previously Presented) The set of instructions of claim 9, wherein said server code and said client code include a number of threads, the method further comprising:

tracking an execution time for each of said threads by a processor in said single computer system; and

tracking a number of transactions completed between the execution of said server code and the execution of said client code wherein said performance data is based on said number of transactions completed over a period of time.

11. (Previously Presented) A set of instructions residing in a storage medium, said set of instructions capable of being executed by a processor to implement a protocol performance test method, the method comprising:

operating a computer system under test as a server and a client to emulate a network configuration;

executing server code on said computer system under test according to a multi-computer communication protocol;

executing client code on said computer system under test according to said multi-computer communication protocol; and

determining performance data for said computer system under test.

12. (Previously Presented) The set of instructions of claim 11 wherein said server code and said client code include a number of threads, the method further comprising:

tracking an execution time for each of said threads by a processor in said computer system under test.

13. (Previously Presented) The set of instructions of claim 12 wherein said multi-computer communication protocol defines transactions between said server and said client, the method further comprising:

tracking a number of transactions completed between the execution of said server code and the execution of said client code.

14. (Original) The set of instructions of claim 13 wherein said performance data is based on said number of transactions completed over a period of time.

15. (Previously Presented) The set of instructions of claim 14 wherein said performance data is based on said number of transactions completed over said period of time modified by a scaling factor.

16. (Previously Presented) The set of instructions of claim 15 wherein said scaling factor comprises a total execution time for both client and server threads divided by one of an execution time for said server threads and an execution time for said client threads.

17. (Previously Presented) A computer system under test to be operated in a multi-computer environment, comprising:
a processor to execute server code and client code on said computer system under test according to a multi-computer communication protocol, said computer system under test to determine performance data for said multi-computer communication protocol, said computer system under test operating as a server and a client to emulate a network configuration.

18. (Previously Presented) The computer system of claim 17 wherein said server code and said client code includes a number of threads, said computer system under test to track an execution time for each of said threads by said processor.

19. (Previously Presented) The computer system of claim 18 wherein said multi-computer communication protocol defines transactions between said server and said client, said computer system under test to track a number of transactions completed between the execution of said server code and the execution of said client code.

20. (Original) The computer system of claim 19 wherein said performance data is based on said number of transactions completed over a period of time.

21. (Previously Presented) The computer system of claim 20 wherein said performance data is based on said number of transactions completed over said period of time modified by a scaling factor.

22. (Previously Presented) The computer system of claim 21 wherein said scaling factor comprises a total execution time for both client and server threads divided by one of an execution time for said server threads and an execution time for said client threads.

23. (Previously Added) The method of claim 1, wherein said server code comprises a number of server threads, said client code comprises a number of client threads, and wherein said method further comprises executing scheduler code, said scheduler code comprising a number of scheduler threads, said scheduler threads for coordinating communication of data between said client threads and said server threads.

-5-

Yeh *et al.*
Appl. No. 09/851,725

24. (Previously Added) The method of claim 23, wherein executing said scheduler code includes interfacing with a queue to store data packets to be transferred to a client thread or a server thread.

25. (Previously Added) The set of instructions of claim 9, wherein said server code comprises a number of server threads, said client code comprises a number of client threads, and wherein said method further comprises executing scheduler code, said scheduler code comprising a number of scheduler threads, said scheduler threads for coordinating communication of data between said client threads and said server threads.

26. (Previously Added) The set of instructions of claim 25, wherein executing said scheduler code includes interfacing with a queue to store data packets to be transferred to a client thread or a server thread.

27. (New) A system for performing protocol performance testing, comprising:
a computer system under test, said computer system having at least one processor and a memory for storing instruction code that when executed by said at least one processor causes said computer system under test to:
emulate a network configuration of a client and a server by executing server threads, client threads, and scheduler threads, said server threads and said client threads to transmit and receive data packets to and from said scheduler threads, said scheduler threads to interface with a queue to temporarily store said data packets, wherein said scheduler threads to coordinate the communication of said data packets between said server threads and said client threads.

28. (New) The system of claim 27, further comprising sockets, wherein said data packets from said server threads and said client threads are transmitted to said sockets and said scheduler threads retrieve said data packets from said sockets to transfer said data packets to the queue.

29. (New) The method of claim 1, wherein the performance data for said single computer system is representative of said single computer system acting as a server or said single computer system acting as a client.

30. (New) The method of claim 3, wherein the performance data for said computer system under test is representative of said computer system under test acting as a server or said single computer system under test acting as a client.